



19/62

ENTITY FSM IS

PORT(
 ports for entity fsm....
);

ARCHITECTURE FSM OF FSM IS

BEGIN

 ... HDL code for FSM and rest of the entity ...

 fsm_state(0 to 2) <= ... Signal 801 ...

```
8 5 3 { --!! Embedded FSM : examplefsm;
8 5 9 { --!! clock      : (fsm_clock);
8 5 4 { --!! state_vector : (fsm_state(0 to 2));
8 5 5 { --!! states      : (S0, S1, S2, S3, S4);
8 5 6 { --!! state_encoding : ('000', '001', '010', '011', '100');
      { --!! arcs        : (S0 => S0, S0 => S1, S0 => S2,
8 5 7 { --!!              (S1 => S2, S1 => S3, S2 => S2,
      { --!!              (S2 => S3, S3 => S4, S4 => S0);
8 5 8 { --!! End FSM;
```

8 5 2 } 8 6 0

END;

Fig. 8C
Prior Art

20/62

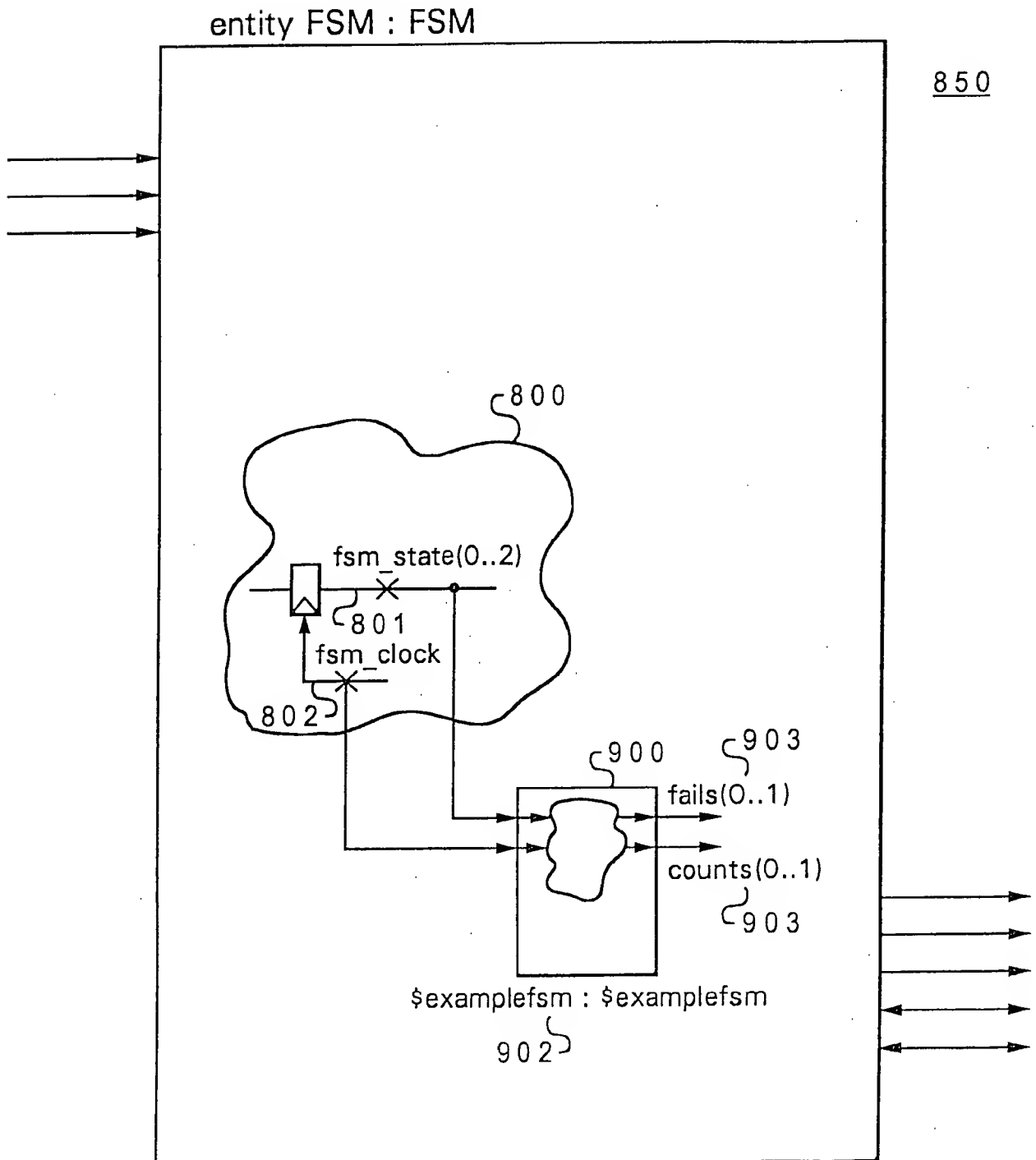
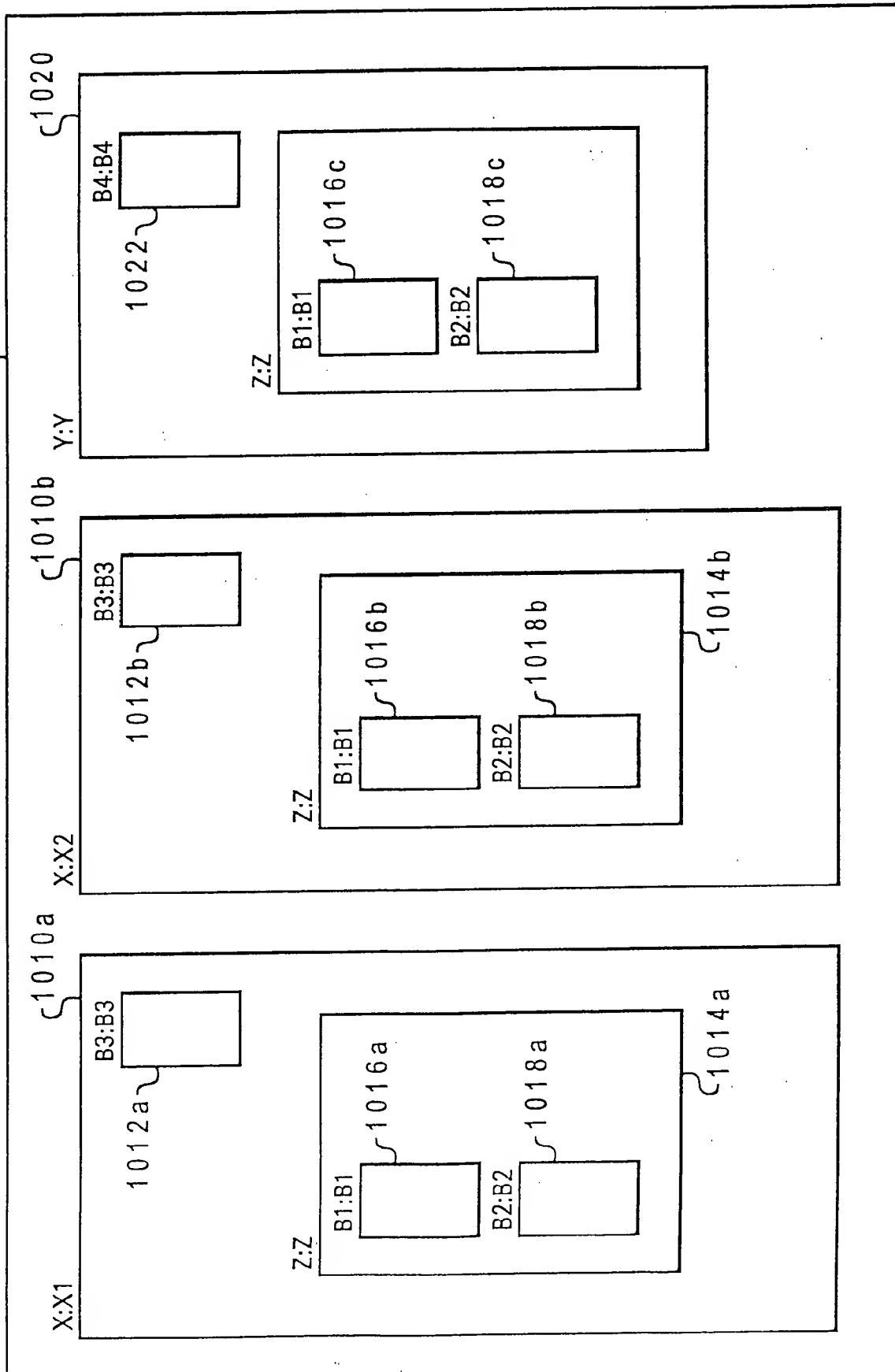


Fig. 9
Prior Art

Fig. 10A Prior Art

1000

TOP:TOP



AUS920000651US1
 Gabele, et al.
 Non-Redundant Collection Of Harvest Events
 Within A Batch Simulation Farm Network

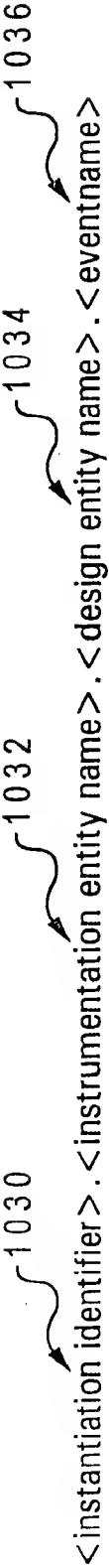


Fig. 10B

X1	B3	X	COUNT1	1040
X1.Z	B1	Z	COUNT1	1041
X1.Z	B2	Z	COUNT1	1042
X2	B3	X	COUNT1	1043
X2.Z	B1	Z	COUNT1	1044
X2.Z	B2	Z	COUNT1	1045
Y	B4	Y	COUNT1	1046
Y.Z	B1	Z	COUNT1	1047
Y.Z	B2	Z	COUNT1	1048

Fig. 10C



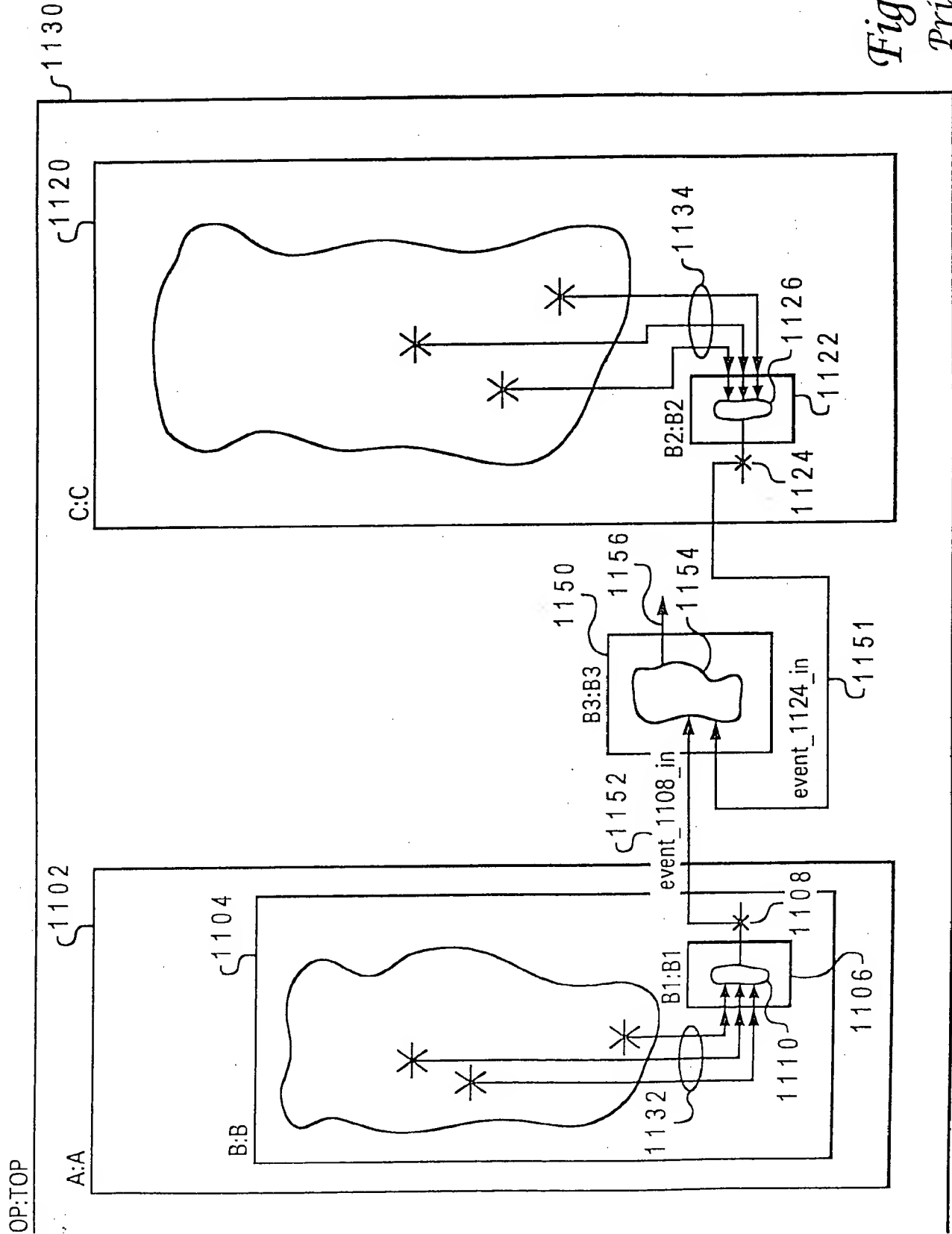
Fig. 10D

Prior Art

AUS920000651US1
 Gabele, et al.
 Non-Redundant Collection Of Harvest Events
 Within A Batch Simulation Farm Network

23/62

Fig. 11A
 Prior Art



24/62

--!! Inputs
--!! event_1108_in <= C.[B2.count.event_1108];
--!! event_1124_in <= A.B.[B1.count.event_1124];
--!! End Inputs

1163 } 1165 } 1161 }
1164 } 1166 } 1162 }

Fig. 11B

--!! Inputs
--!! event_1108_in <= C.[count.event_1108];
--!! event_1124_in <= B.[count.event_1124];
--!! End Inputs

1171 }
1172 }

Fig. 11C

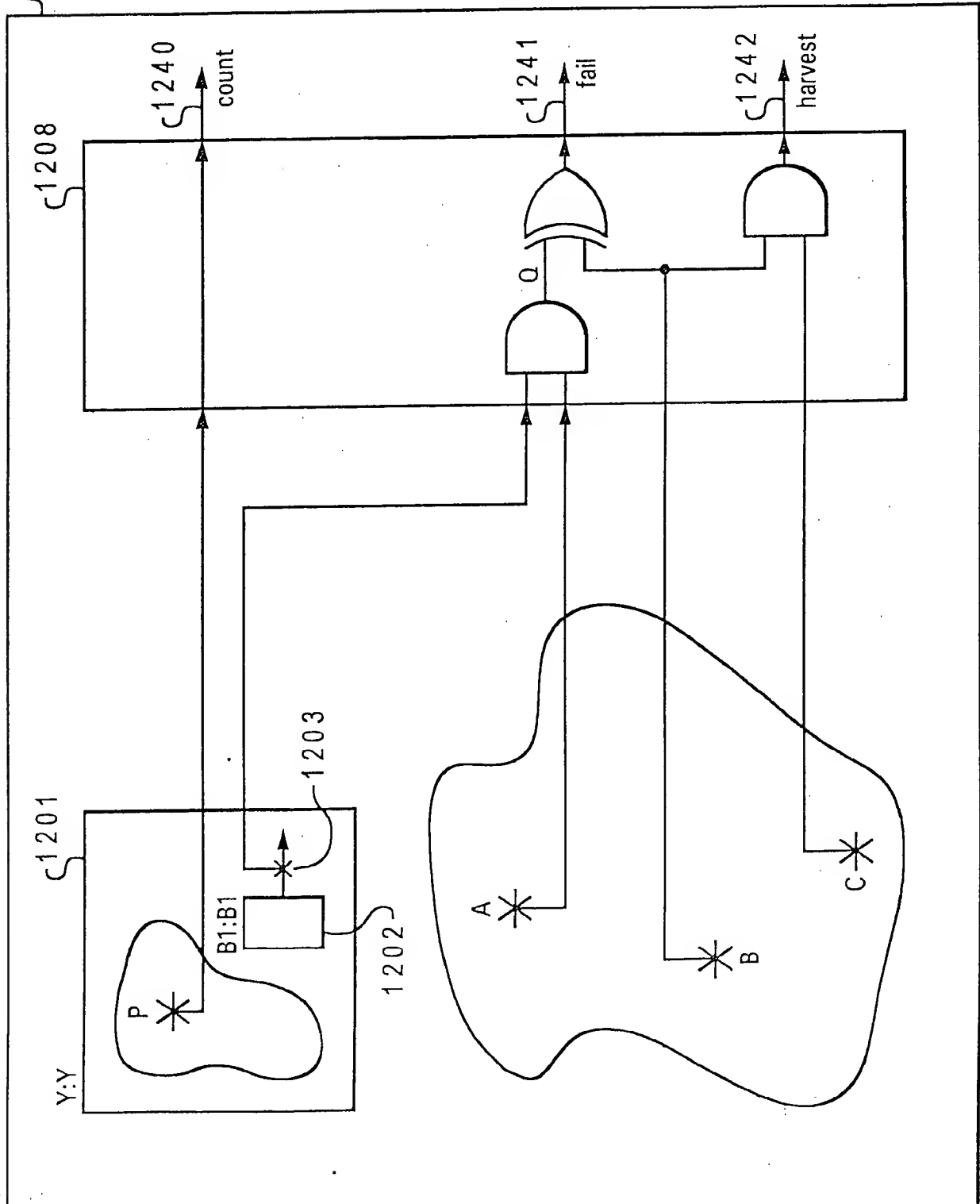
Prior Art

AUS920000651US1
 Gabele, et al.
 Non-Redundant Collection Of Harvest Events
 Within A Batch Simulation Farm Network

25/62

Fig. 12A
 Prior Art

1200



```

ENTITY X IS
    PORT(      :
              :
              :
           );

ARCHITECTURE example OF X IS
BEGIN
    .
    .
    .
    ... HDL code for X ...
    .
    .
    .
    .
    .
    Y:Y
    PORT MAP(   :
               :
               :
            );

    A <= ....
    B <= ....
    C <= ....

    --!! [count, countname0, clock] <= Y.P;
    --!! Q <= Y. [B1.count.count1] AND A;
    --!! [fail, failname0, "fail msg"] <= Q XOR B;
    --!! [harvest, harvestname0, "harvest msg"] <= B AND C;

END;
```

Fig. 12B
Prior Art

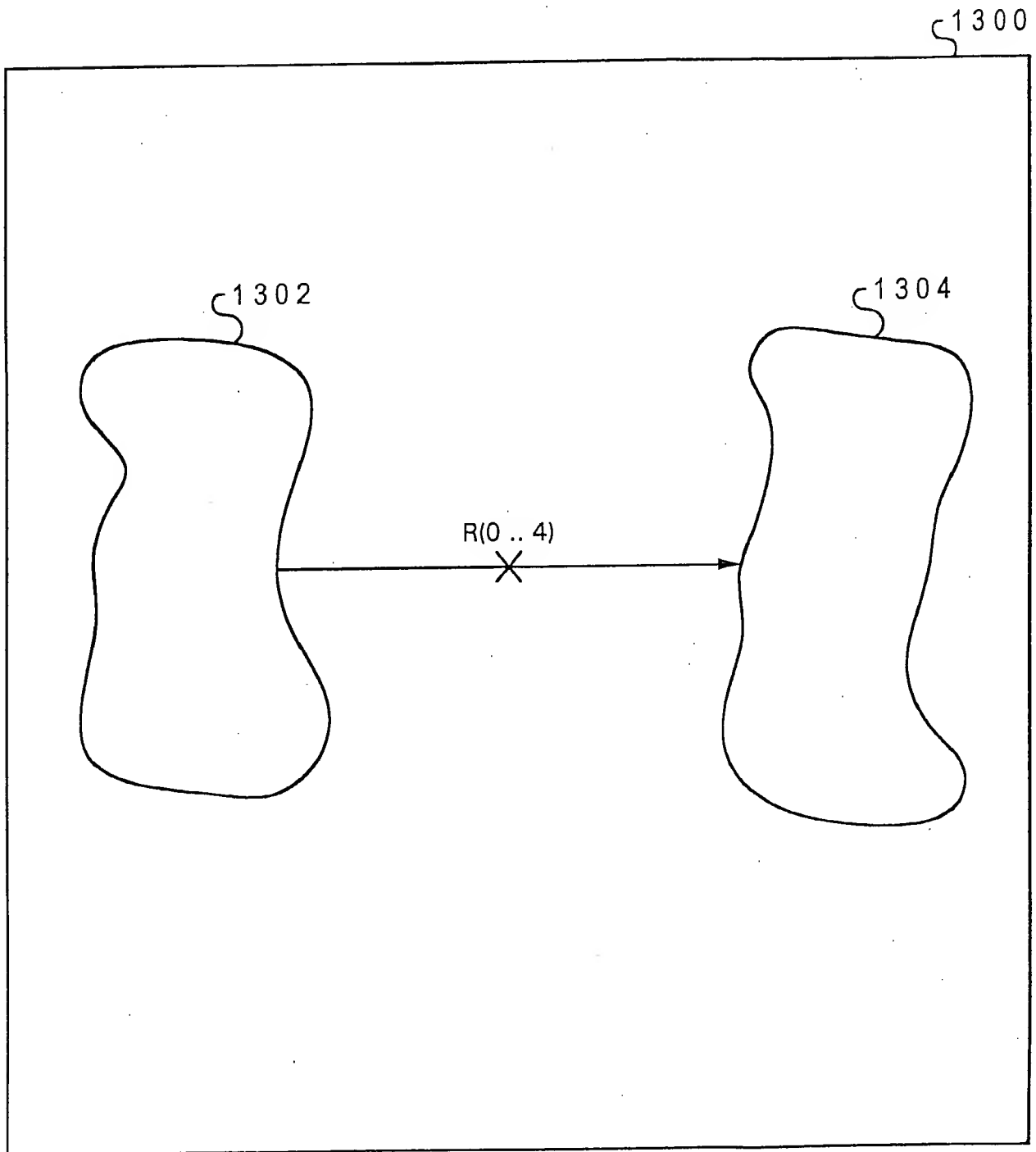


Fig. 13A
Prior Art

28/62

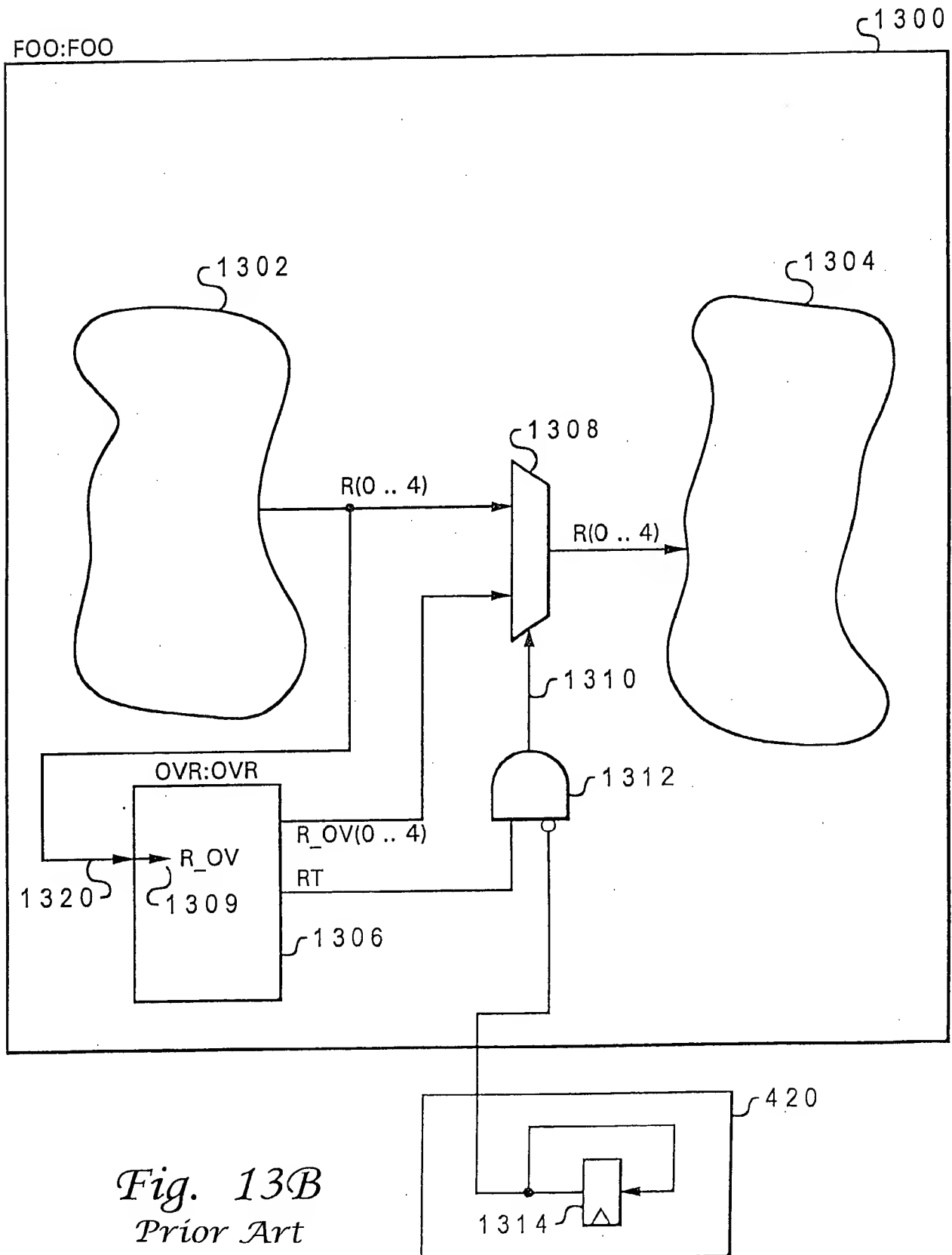


Fig. 13B
Prior Art

AUS920000651US1

Gabele, et al.

Non-Redundant Collection Of Harvest Events
Within A Batch Simulation Farm Network

29/62

```

ENTITY OVR IS
    PORT(
        R_IN      : IN std_ulogic_vector(0 .. 4);
        .
        .
        ... other ports as required ...
        .
        .
        R_OV      : OUT std_ulogic_vector(0 .. 4);
        RT        : OUT std_ulogic
    );

--!! BEGIN
--!! Design Entity: FOO;

--!! Inputs (0 to 4)
--!! R_IN => {R(0 .. 4)};
--!! :
--!! ... other ports as needed ...
--!! :
--!! End Inputs

--!! Outputs
--!! <R_OVRRIDE> : R_OV(0 .. 4) => R(0 .. 4) [RT];
--!! End Outputs

--!! End

ARCHITECTURE example of OVR IS
BEGIN
    ... HDL code for entity body section ...
END;

```

Handwritten annotations in the figure include curly braces and numbers grouping specific lines of code:

- A brace labeled **1 3 6 4** groups the `R_IN` input port declaration.
- A brace labeled **1 3 6 2** groups the `R_OV` and `RT` output port declarations.
- A brace labeled **1 3 6 3** groups the `--!! End Inputs` and `--!! Outputs` comments.
- A brace labeled **1 3 6 1** groups the `--!! <R_OVRRIDE> : R_OV(0 .. 4) => R(0 .. 4) [RT];` line.
- A brace labeled **1 3 6 0** groups the `--!! R_IN => {R(0 .. 4)};` line.
- A brace labeled **1 3 5 1** groups the `--!! End Inputs` and `--!! Outputs` comments.
- A brace labeled **1 3 5 6** groups the `--!! Outputs` and `--!! End Outputs` comments.
- A brace labeled **1 3 5 8** groups the `... HDL code for entity body section ...` line.
- A large brace labeled **1 3 4 0** groups the entire `ENTITY OVR IS` block.

Fig. 13C
Prior Art

AUS920000651US1
 Gabele, et al.
 Non-Redundant Collection Of Harvest Events
 Within A Batch Simulation Farm Network

30/62

ENTITY FOO IS

```

    PORT(      :
              :
              :
    );
  
```

ARCHITECTURE example of FOO IS

BEGIN

```

    .
    .
    .
    .
    R <= .....
    .
    .
    .
    .
    {
      --!! R_IN <= {R};
      --!!
      --!!
      --!! R OV(0 to 4) <= .....;
      --!! RT <= .....;
      --!! [override, R_OVRRIDE, R(0 .. 4), RT] <= R_OV(0 to 4);
    }
  
```

1380 {

1381 {

1382 {

1383 {

1384 {

*Fig. 13D**Prior Art*

Fig. 14A
Prior Art

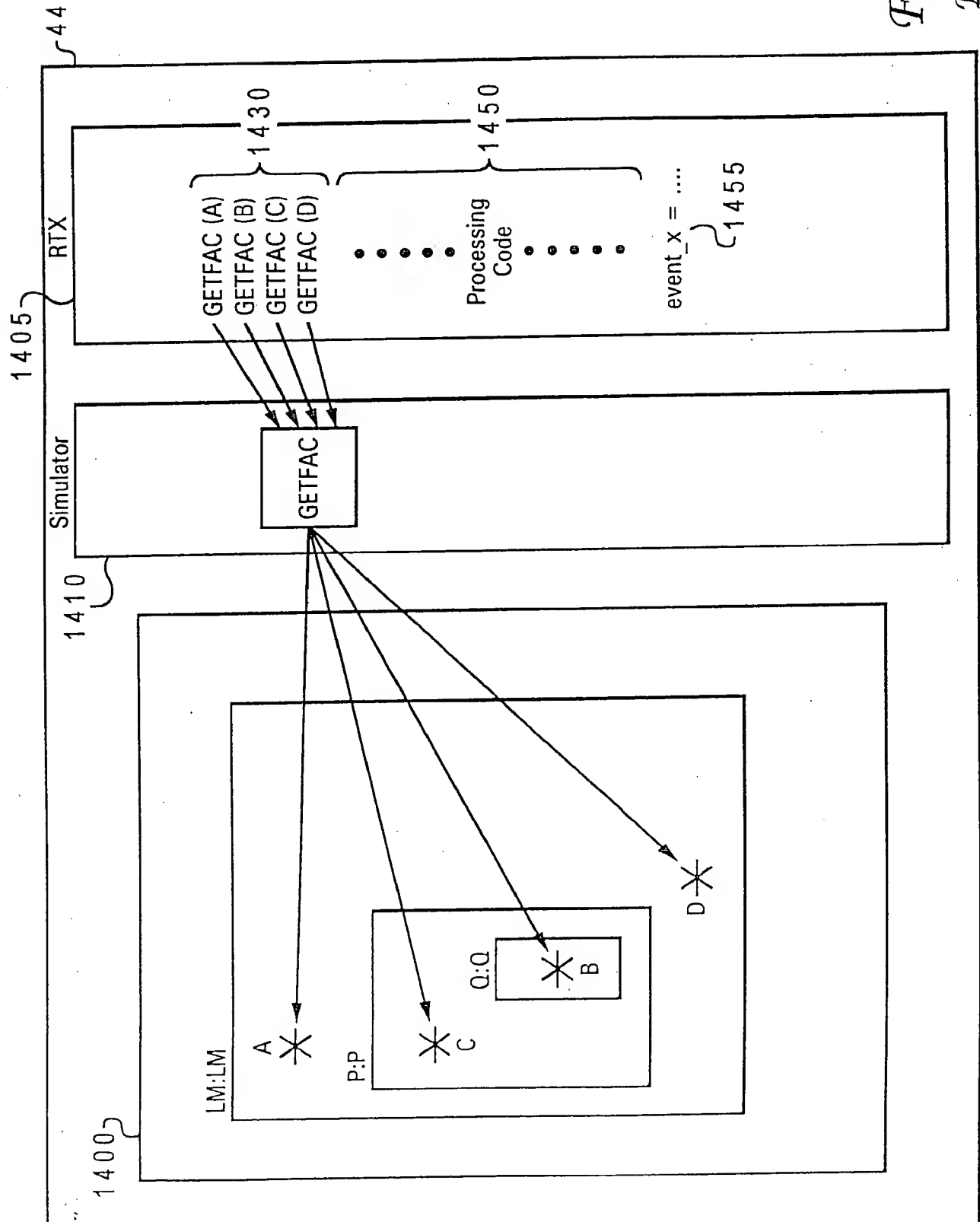
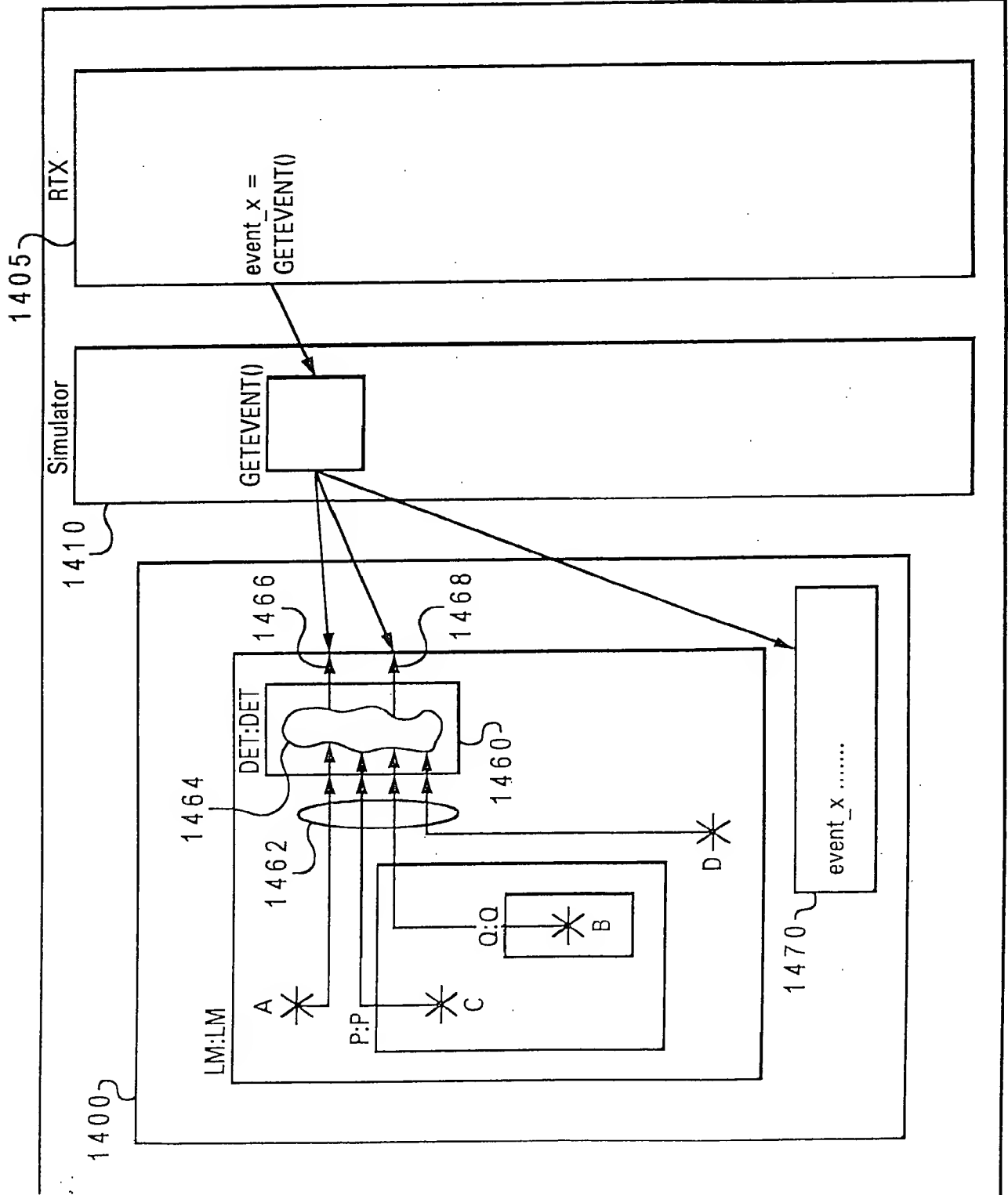


Fig. 14B
Prior Art



AUS920000651US1
 Gabele, et al.
 Non-Redundant Collection Of Harvest Events
 Within A Batch Simulation Farm Network

33/62

```

ENTITY DET IS
    PORT(
        A      : IN std_ulogic;
        B      : IN std_ulogic_vector(0 to 5);
        C      : IN std_ulogic;
        D      : IN std_ulogic;
        :
        :
        event_x : OUT std_ulogic_vector(0 to 2);
        x_here  : OUT std_ulogic;
    );

    --!! BEGIN
    --!! Design Entity: LM;

    --!! Inputs
    --!! A  => A;
    --!! B  => P.Q.B;
    --!! C  => P.C;
    --!! D  => D;
    --!! End Inputs

    --!! Detections
    --!! <event_x>:event_x(0 to 2) [x_here];
    --!! End Detections

    --!! End;

    ARCHITECTURE example of DET IS
    BEGIN
        ... HDL code ...

    END;
  
```

1491 {

1492 {

1493 {

1494 {

1495 {

1480 }

Fig. 14C
Prior Art